#3

02985.000374

PATENT APPLICATION

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:

NAOYUKI OE ET AL.

Application No.: 09/988,106

Filed: November 19, 2001

For: INFORMATION PROCESSING METHOD, APPARATUS, AND SYSTEM FOR CONTROLLING COMPUTER RESOURCES, CONTROL METHOD THEREFOR, STORAGE MEDIUM, AND PROGRAM

Examiner: Not Yet Assigned

Group Art Unit: 2152

January 15, 2002

Commissioner for Patents
Washington, D.C. 20231

CLAIM TO PRIORITY

Sir:

Applicants hereby claim priority under the International Convention and all rights to which he is entitled under 35 U.S.C. § 119 based upon the following Japanese Priority Applications:

2000-352113 filed on November 20, 2000

2001-161403 filed on April 23, 2001

2001-190445 filed on May 22, 2001

2001-322437 filed on October 19, 2001

Certified copies of the priority documents are enclosed.

Applicants' undersigned attorney may be reached in our New York office by telephone at (212) 218-2100. All correspondence should continue to be directed to our address given below.

Respectfully submitted,

Joseph W. Ragusa

Attorney for Applicants

Registration No. 38,586

FITZPATRICK, CELLA, HARPER & SCINTO
30 Rockefeller Plaza
New York, New York 10112-3801
Facsimile: (212) 218-2200

NY_MAIN 231323 v 1

(Translation of the front page
of the priority document of
Japanese Patent Application
No. 2000-352113)

PATENT OFFICE
JAPANESE GOVERNMENT

This is to certify that the annexed is a true copy of the following application as filed with this Office.

Date of Application : November 20, 2000

Application Number : Patent Application
2000-352113

Applicant(s) : Humming Heads Inc.

November 9, 2001

Commissioner,
Japan Patent Office Kouzo Oikawa

Certification Number 2001-3098287

日 本 国 特 許 庁
JAPAN PATENT OFFICE

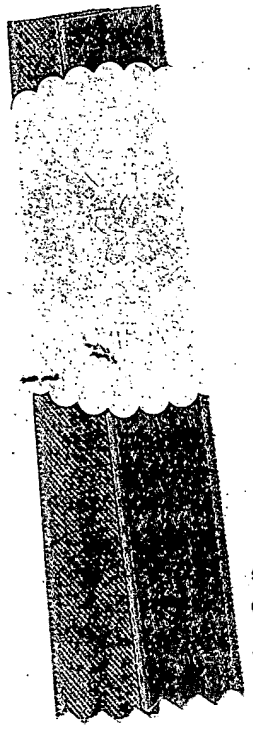
別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出願年月日
Date of Application: ２０００年１１月２０日

出願番号
Application Number: 特願２０００－３５２１１３

出願人
Applicant(s): ハミングヘッズ株式会社



２００１年１１月 ９日

特許庁長官
Commissioner,
Japan Patent Office

及川耕造

出証番号 出証特２００１－３０９８２８７

【書類名】　特許願
【整理番号】　HH00001
【提出日】　平成12年11月20日
【あて先】　特許庁長官殿
【国際特許分類】　G06F 12/00
【発明者】
　【住所又は居所】　東京都中央区月島一丁目２番１３号　ハミングヘッズ株式会社内
　【氏名】　大江　尚之
【発明者】
　【住所又は居所】　東京都中央区月島一丁目２番１３号　ハミングヘッズ株式会社内
　【氏名】　志摩　貴浩
【特許出願人】
　【識別番号】　500083226
　【氏名又は名称】　ハミングヘッズ株式会社
【代理人】
　【識別番号】　100088720
　【弁理士】
　【氏名又は名称】　小川　眞一
　【電話番号】　03-3256-8439
【手数料の表示】
　【予納台帳番号】　052504
　【納付金額】　21,000円
【提出物件の目録】
　【物件名】　明細書　1
　【物件名】　図面　1
　【物件名】　要約書　1
【プルーフの要否】　要

【書類名】　明細書

【発明の名称】　コンピュータリソースの制御方法および装置並びに記録媒体

【特許請求の範囲】

【請求項１】　ファイル、ネットワーク、記憶装置、表示画面、外部付属装置等のオペレーティングシステムが管理しているコンピュータリソースに対するアクセスを制御する方法であって、

前記コンピュータリソースに対するプロセスまたはオペレーティングシステムからの操作要求をコンピュータリソースにアクセスする前に捕捉する第１のステップと、

前記第１のステップで捕捉した操作要求によって指定されるコンピュータリソースに対するアクセス権限があるか否かを判定する第２のステップと、

アクセス権限があれば当該操作要求通りにオペレーティングシステムに渡し、その結果を要求元プロセスに返す第３のステップと、

アクセス権限がなければ当該操作要求を拒否する第４のステップと

を備えることを特徴とするコンピュータリソースの制御方法。

【請求項２】　前記第１のステップに代えて、前記コンピュータリソースに対するプロセス及びオペレーティングシステムからの操作要求をコンピュータリソースにアクセスする前に捕捉する第５のステップを備えることを特徴とする請求項１に記載のコンピュータリソースの制御方法。

【請求項３】　前記第２のステップが、

特定のコンピュータリソースを指定するリソース指定情報、アクセス権が有効となる条件情報、既存環境で定義されていない拡張したアクセス権を指定するアクセス権限情報を含むアクセス権管理テーブルを参照し、アクセス権限があるか否かを判定するステップまたは、

コンピュータリソース内部に記述された、既存環境で定義されていない拡張したアクセス権を指定するアクセス権限情報を参照し、アクセス権限があるか否かを判定するステップまたは、

アクセス権限が獲得できたか否かをもって、アクセス権限があるか否かを判定するステップ

を備えることを特徴とする請求項１または２に記載のコンピュータリソースの制御方法。

【請求項４】　前記アクセス権限情報は、他媒体への移動権限、他媒体へのコピー権限、印刷権限、共有メモリへの読み込み権限、画面ハードコピー権限、使用プロセスの限定権限のうち少なくとも１つを指定する情報を含むことを特徴とする請求項３に記載のコンピュータリソースの制御方法。

【請求項５】　前記第４のステップは、

要求されたコンピュータリソースにアクセスせずにアクセス違反のエラー通知を要求元プロセスに返す、または要求されたコンピュータリソースにアクセスせずにアクセス成功の通知を要求元プロセスに返す、またはダミーのコンピュータリソースへの操作要求に変換してオペレーティングシステムに渡し、その結果を要求元プロセスに返すステップから成ることを特徴とする請求項１～４のいずれか一項に記載のコンピュータリソースの制御方法。

【請求項６】　ファイル、ネットワーク、記憶装置、表示画面、外部付属装置等のオペレーティングシステムが管理しているコンピュータリソースに対するアクセスを制御するリソース制御手段を備えたコンピュータ装置であって、

前記リソース制御手段が、

前記コンピュータリソースに対するプロセスまたはオペレーティングシステムからの操作要求をコンピュータリソースにアクセスする前に捕捉する第１の手段と、

前記第１の手段で捕捉した操作要求によって指定されるコンピュータリソースに対するアクセス権限があるか否かを判定する第２の手段と、

アクセス権限があれば当該操作要求通りにオペレーティングシステムに渡し、その結果を要求元プロセスに返す第３の手段と、

アクセス権限がなければ当該操作要求を拒否する第４の手段と

を備えることを特徴とするコンピュータ装置。

【請求項７】　前記第１の手段に代えて、前記コンピュータリソースに対するプロセス及びオペレーティングシステムからの操作要求をコンピュータリソースにアクセスする前に捕捉する第５の手段を備えることを特徴とする請求項６に

記載のコンピュータ装置。

【請求項８】　前記第２の手段が、

特定のコンピュータリソースを指定するリソース指定情報、アクセス権が有効となる条件情報、既存環境で定義されていない拡張したアクセス権を指定するアクセス権限情報を含むアクセス権管理テーブルを参照し、アクセス権限があるか否かを判定する手段または、

コンピュータリソース内部に記述された、既存環境で定義されていない拡張したアクセス権を指定するアクセス権限情報を参照し、アクセス権限があるか否かを判定する手段または、

アクセス権限が獲得できたか否かをもって、アクセス権限があるか否かを判定する手段

を備えることを特徴とする請求項６または７に記載のコンピュータ装置。

【請求項９】　前記アクセス権限情報は、他媒体への移動権限、他媒体へのコピー権限、印刷権限、共有メモリへの読み込み権限、画面ハードコピー権限、使用プロセスの限定権限のうち少なくとも１つを指定する情報を含むことを特徴とする請求項８に記載のコンピュータ装置。

【請求項１０】　前記第４の手段は、

要求されたコンピュータリソースにアクセスせずにアクセス違反のエラー通知を要求元プロセスに返す、または要求されたコンピュータリソースにアクセスせずにアクセス成功の通知を要求元プロセスに返す、またはダミーのコンピュータリソースへの操作要求に変換してオペレーティングシステムに渡し、その結果を要求元プロセスに返す手段から成ることを特徴とする請求項６～９のいずれか一項に記載のコンピュータ装置。

【請求項１１】　ファイル、ネットワーク、記憶装置、表示画面、外部付属装置等のコンピュータリソースに対するアクセスを制御するリソース制御プログラムを記録した媒体であって、

前記リソース制御プログラムが、

前記コンピュータリソースに対するプロセスまたはオペレーティングシステムからの操作要求をコンピュータリソースにアクセスする前に捕捉する第１の処理

と、

前記第１の処理で捕捉した操作要求によって指定されるコンピュータリソースに対するアクセス権限があるか否かを判定する第２の処理と、

アクセス権限があれば当該操作要求通りにオペレーティングシステムに渡し、その結果を要求元プロセスに返す第３の処理と、

アクセス権限がなければ当該操作要求を拒否する第４の処理と

を備えることを特徴とするコンピュータが読取り可能なプログラムを記録した記録媒体。

【請求項１２】　前記第１の処理に代えて、前記コンピュータリソースに対するプロセス及びオペレーティングシステムからの操作要求をコンピュータリソースにアクセスする前に捕捉する第５の処理を備えることを特徴とする請求項１１に記載の記録媒体。

【請求項１３】　前記第２の処理が、

特定のコンピュータリソースを指定するリソース指定情報、アクセス権が有効となる条件情報、既存環境で定義されていない拡張したアクセス権を指定するアクセス権限情報を含むアクセス権管理テーブルを参照し、アクセス権限があるか否かを判定する処理または、

コンピュータリソース内部に記述された、既存環境で定義されていない拡張したアクセス権を指定するアクセス権限情報を参照し、アクセス権限があるか否かを判定する処理または、

アクセス権限が獲得できたか否かをもって、アクセス権限があるか否かを判定する処理

を備えることを特徴とする請求項１１または１２に記載の記録媒体。

【請求項１４】　前記アクセス権限情報は、他媒体への移動権限、他媒体へのコピー権限、印刷権限、共有メモリへの読み込み権限、画面ハードコピー権限、使用プロセスの限定権限のうち少なくとも１つを指定する情報を含むことを特徴とする請求項１３に記載の記録媒体。

【請求項１５】　前記第４の処理は、

要求されたコンピュータリソースにアクセスせずにアクセス違反のエラー通知

を要求元プロセスに返す、または要求されたコンピュータリソースにアクセスせずにアクセス成功の通知を要求元プロセスに返す、またはダミーのコンピュータリソースへの操作要求に変換してオペレーティングシステムに渡し、その結果を要求元プロセスに返す処理から成ることを特徴とする請求項１１～１４のいずれか一項に記載の記録媒体。

【請求項１６】　ファイル、ネットワーク、記憶装置、表示画面、外部付属装置等のコンピュータリソースに対するアクセスを制御するリソース制御プログラムであって、

前記コンピュータリソースに対するプロセスまたはオペレーティングシステムからの操作要求をコンピュータリソースにアクセスする前に捕捉する第１の処理と、

前記第１の処理で捕捉した操作要求によって指定されるコンピュータリソースに対するアクセス権限があるか否かを判定する第２の処理と、

アクセス権限があれば当該操作要求通りにオペレーティングシステムに渡し、その結果を要求元プロセスに返す第３の処理と、

アクセス権限がなければ当該操作要求を拒否する第４の処理と

を備えることを特徴とするコンピュータが実行可能なリソース制御プログラム。

【請求項１７】　前記第１の処理に代えて、前記コンピュータリソースに対するプロセス及びオペレーティングシステムからの操作要求をコンピュータリソースにアクセスする前に捕捉する第５の処理を備えることを特徴とする請求項１６に記載のリソース制御プログラム。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】

本発明は、ファイル、記憶装置、表示画面、外部付属装置等のコンピュータリソースに対するアクセスを管理するコンピュータリソースの制御方法および装置並びに記録媒体に関するものである。

【０００２】

【従来の技術】

従来において、パーソナルコンピュータ等のコンピュータにおけるファイルや記憶装置等のリソースをアプリケーションプログラムを介してユーザがアクセスする場合に、アクセス権限のないユーザに情報が解読または盗聴されるのを防ぐために、オペレーティングシステム（以下、ＯＳ）内にアクセス権限のチェック機能を設ける方法、あるいは専用のアクセス管理ツールを付加してアクセス権限のチェックを行なう方法が知られている。

【０００３】

例えばWindows（米国マイクロソフト社の登録商標）に代表される汎用のＯＳにおいては、ファイルの読取り、書き込み、実行をアクセス権限のないユーザに対しては許可しない機能が備わっている。また、ファイルの削除、アクセス権限の変更、所有権の変更についての権限を設定可能にした汎用ＯＳもある。

また、アクセス管理ツールとして、例えば特開平７－８４８５２公報に開示されているように、ファイルの参照と共に複写の可否を登録し、その可否によって参照、複写を制限するものが知られている。詳しくは、表示領域に読出し制限の属性を付加し、表示画面のハードコピーを防止するものが知られている。

【０００４】

【発明が解決しようとする課題】

アクセス権限のないユーザに対して情報の持ち出しを全面的に禁止するためには、図９に示すように、メールへの添付、印刷、ファイル移動／ファイルコピー、クリップボードへのコピー、フロッピーディスクへの別名保存、オブジェクトの貼り付け、キャプチャーなどの機能を制限する必要がある。さらに、ネットワークを通じた情報の持ち出しを制限する必要がある。

しかしながら、上記従来技術にあっては、ファイル及び画面のハードコピー以外の操作（例えばクリップボードへのコピー）に対して制限することができないという問題がある。もしも、クリップボードへのコピーなどの操作を制限しようとする場合には、ＯＳまたはアプリケーション自体に変更を加えることが必要になり、汎用的な応用ができないという問題がある。

【０００５】

本発明の目的は、ＯＳやプロセス（ＯＳの元に稼動しているプログラムであり

、アプリケーションやデーモンなど）を変更することなく、ファイルや画面以外のコンピュータリソースを含めてアクセス権限のないユーザに対するリソースの操作を制限し、しかも既存環境における禁止または制限事項を拡張することができるコンピュータリソースの制御方法および装置並びに記録媒体を提供することにある。

【０００６】

【課題を解決するための手段】

上記目的を達成するために、本発明のコンピュータリソースの制御方法は、ファイル、ネットワーク、記憶装置、表示画面、外部付属装置等のオペレーティングシステムが管理しているコンピュータリソースに対するプロセスまたはオペレーティングシステムからの操作要求をコンピュータリソースにアクセスする前に捕捉する第１のステップと、前記第１のステップで捕捉した操作要求によって指定されるコンピュータリソースに対するアクセス権限があるか否かを判定する第２のステップと、アクセス権限があれば当該操作要求通りにオペレーティングシステムに渡し、その結果を要求元プロセスに返す第３のステップと、アクセス権限がなければ当該操作要求を拒否する第４のステップとを備えることを特徴とする。

また、前記第１のステップに代えて、前記コンピュータリソースに対するプロセス及びオペレーティングシステムからの操作要求をコンピュータリソースにアクセスする前に捕捉する第５のステップを備えることを特徴とする。

また、前記第２のステップが、特定のコンピュータリソースを指定するリソース指定情報、アクセス権が有効となる条件情報、既存環境で定義されていない拡張したアクセス権を指定するアクセス権限情報を含むアクセス権管理テーブルを参照し、アクセス権限があるか否かを判定するステップ、またはコンピュータリソース内部に記述された既存環境で定義されていない拡張したアクセス権を指定するアクセス権限情報を参照し、アクセス権限があるか否かを判定するステップ、またはアクセス権限が獲得できたか否かをもって、アクセス権限があるか否かを判定するステップを備えることを特徴とする。

また、前記アクセス権限情報は、他媒体への移動権限、他媒体へのコピー権限

、印刷権限、共有メモリへの読み込み権限、画面ハードコピー権限、使用プロセスの限定権限のうち少なくとも１つを指定する情報を含むことを特徴とする。

また、前記第４のステップは、要求されたコンピュータリソースにアクセスせずにアクセス違反のエラー通知を要求元プロセスに返す、または要求されたコンピュータリソースにアクセスせずにアクセス成功の通知を要求元プロセスに返す、またはダミーのコンピュータリソースへの操作要求に変換してオペレーティングシステムに渡し、その結果を要求元プロセスに返すステップから成ることを特徴とする。

【０００７】

さらに、コンピュータリソースに対するアクセスを制御するリソース制御手段を備えた本発明に係るコンピュータ装置は、

前記リソース制御手段が、

前記コンピュータリソースに対するプロセスまたはオペレーティングシステムからの操作要求をコンピュータリソースにアクセスする前に捕捉する第１の手段と、前記第１の手段で捕捉した操作要求によって指定されるコンピュータリソースに対するアクセス権限があるか否かを判定する第２の手段と、アクセス権限があれば当該操作要求通りにオペレーティングシステムに渡し、その結果を要求元プロセスに返す第３の手段と、アクセス権限がなければ当該操作要求を拒否する第４の手段とを備えることを特徴とする。

また、前記第１の手段に代えて、前記コンピュータリソースに対するプロセス及びオペレーティングシステムからの操作要求をコンピュータリソースにアクセスする前に捕捉する第５の手段を備えることを特徴とする。

また、前記第２の手段が、特定のコンピュータリソースを指定するリソース指定情報、アクセス権が有効となる条件情報、既存環境で定義されていない拡張したアクセス権を指定するアクセス権限情報を含むアクセス権管理テーブルを参照し、アクセス権限があるか否かを判定する手段、またはコンピュータリソース内部に記述された、既存環境で定義されていない拡張したアクセス権を指定するアクセス権限情報を参照し、アクセス権限があるか否かを判定する手段、またはアクセス権限が獲得できたか否かをもって、アクセス権限があるか否かを判定する

手段を備えることを特徴とする。

また、前記アクセス権限情報は、他媒体への移動権限、他媒体へのコピー権限、印刷権限、共有メモリへの読み込み権限、画面ハードコピー権限、使用プロセスの限定権限のうち少なくとも１つを指定する情報を含むことを特徴とする。

また、前記第４の手段は、要求されたコンピュータリソースにアクセスせずにアクセス違反のエラー通知を要求元プロセスに返す、または要求されたコンピュータリソースにアクセスせずにアクセス成功の通知を要求元プロセスに返す、またはダミーのコンピュータリソースへの操作要求に変換してオペレーティングシステムに渡し、その結果を要求元プロセスに返す手段から成ることを特徴とする。

【０００８】

さらに本発明に係るリソース制御プログラムを記録した媒体は、前記リソース制御プログラムが、

前記コンピュータリソースに対するプロセスまたはオペレーティングシステムからの操作要求をコンピュータリソースにアクセスする前に捕捉する第１の処理と、前記第１の処理で捕捉した操作要求によって指定されるコンピュータリソースに対するアクセス権限があるか否かを判定する第２の処理と、アクセス権限があれば当該操作要求通りにオペレーティングシステムに渡し、その結果を要求元プロセスに返す第３の処理と、アクセス権限がなければ当該操作要求を拒否する第４の処理とを備えることを特徴とする。

また、前記第１の処理に代えて、前記コンピュータリソースに対するプロセス及びオペレーティングシステムからの操作要求をコンピュータリソースにアクセスする前に捕捉する第５の処理を備えること特徴とする。

また、前記第２の処理が、特定のコンピュータリソースを指定するリソース指定情報、アクセス権が有効となる条件情報、既存環境で定義されていない拡張したアクセス権を指定するアクセス権限情報を含むアクセス権管理テーブルを参照し、アクセス権限があるか否かを判定する処理、またはコンピュータリソース内部に記述された、既存環境で定義されていない拡張したアクセス権を指定するアクセス権限情報を参照し、アクセス権限があるか否かを判定する処理、またはア

クセス権限が獲得できたか否かをもって、アクセス権限があるか否かを判定する処理

を備えることを特徴とする。

また、前記アクセス権限情報は、他媒体への移動権限、他媒体へのコピー権限、印刷権限、共有メモリへの読み込み権限、画面ハードコピー権限、使用プロセスの限定権限のうち少なくとも１つを指定する情報を含むことを特徴とする。

また、前記第４の処理は、要求されたコンピュータリソースにアクセスせずにアクセス違反のエラー通知を要求元プロセスに返す、または要求されたコンピュータリソースにアクセスせずにアクセス成功の通知を要求元プロセスに返す、またはダミーのコンピュータリソースへの操作要求に変換してオペレーティングシステムに渡し、その結果を要求元プロセスに返す処理から成ることを特徴とする。

【０００９】

さらにコンピュータリソースに対するアクセスを制御する本発明に係るリソース制御プログラムは、前記コンピュータリソースに対するプロセスまたはオペレーティングシステムからの操作要求をコンピュータリソースにアクセスする前に捕捉する第１の処理と、前記第１の処理で捕捉した操作要求によって指定されるコンピュータリソースに対するアクセス権限があるか否かを判定する第２の処理と、アクセス権限があれば当該操作要求通りにオペレーティングシステムに渡し、その結果を要求元プロセスに返す第３の処理と、アクセス権限がなければ当該操作要求を拒否する第４の処理とを備えることを特徴とする。

また、前記第１の処理に代えて、前記コンピュータリソースに対するプロセス及びオペレーティングシステムからの操作要求をコンピュータリソースにアクセスする前に捕捉する第５の処理を備えることを特徴とする。

【００１０】

【発明の実施の形態】

以下、本発明の実施の形態を図面により詳細に説明する。

図１（ａ），（ｂ）は本発明を実施する環境の一実施の形態を示すハードウェア構成図である。

図１（ａ）に示す構成は、スタンドアロン構成におけるコンピュータ１０１のハード構成を示すものであり、ハードディスク（ＨＤＤ）１０１１を備えたパーソナルコンピュータ（ＰＣ）１０１２、ディスプレイ１０１３、プリンタ１０１４、外部にリソースデータを出力することが可能な外部装置１０１５で構成されている。

パーソナルコンピュータ１０１２には、汎用のＯＳとアプリケーションが組み込まれており、さらに本発明に係るリソース管理プログラムが組み込まれている。

図１（ｂ）は、ネットワーク１０２を利用する場合の構成を示すものであり、図１（ａ）に示したのと同様な構成のコンピュータ１０１ａ～１０１ｃがネットワーク１０２で接続されている。

【００１１】

このような構成において、一般的に、アプリケーションがＯＳの管理するリソースにアクセスするには、ＯＳが提供するＡＰＩ（Application Program Interface））を利用する。このＡＰＩの利用方法はＯＳにより確定しており、ＡＰＩを利用する実行コード部を判別することができる。本発明では、リソースへのアクセスに必要なすべてのＡＰＩを監視する監視ルーチンを設け、アプリケーションがＡＰＩを利用する前に、その実行コード部を変更するか、ＡＰＩ処理の入り口を監視ルーチンと置き換えることで、ＡＰＩ利用時に監視ルーチンが利用されるようにする。監視ルーチンは、アプリケーションが求めるＡＰＩを処理するか、もしくはＡＰＩの処理をせずに不正命令としてアプリケーションに結果を返す。本発明のリソース管理プログラムによって拡張したアクセス権の管理は、ＯＳの管理とは別に本プログラムが管理し、アクセス権の種類別に監視ルーチンを設ける。この方法により、リソースを不正に利用するアプリケーションから、そのアクセスを制限する。

【００１２】

図２は、本発明に係るリソース管理プログラム２０３の構成及びＡＰＩ監視／制御の概念を示す図であり、リソース管理プログラム２０３はＡＰＩ監視コントローラ（ＡＰＩ監視ＣＴＲＬ）２０３１、ＡＰＬ監視コントローラ（ＡＰＬ監視

ＣＴＲＬ）２０３２、アクセス制御コントローラ（アクセス制御ＣＴＲＬ）２０３３、ＯＳ監視コントローラ（ＯＳ監視ＣＴＲＬ）２０３４から構成されており、リソースアクセス要求を出すアプリケーション２０２１や画面キャプチャなどのＯＳ機能操作２０２２を備える一般的なアプリケーションからなるユーザ環境２０２と汎用ＯＳ２０１との間に位置し、汎用ＯＳ２０１およびユーザ環境２０２が提供するリソースに対する要求を監視するようになっている。

なお、汎用ＯＳ２０１は、ＯＳが管理するリソース２０１１と、ＯＳがアプリケーション２０２１に提供しているＡＰＩ群２０１２を備える。

【００１３】

本発明に係るリソース管理プログラム２０３におけるＡＰＩ監視ＣＴＲＬ２０３１は、アクセス制御を行なうのに必要な全てのＡＰＩを監視するモジュールである。また、ＡＰＬ監視ＣＴＲＬ２０３２は、アプリケーショ２０２１ンが保持しているリソースを記憶するモジュールである。アクセス制御ＣＴＲＬ２０３３はアクセスが許可されているかを判断するモジュールであり、アクセス権管理テーブル２０３５を備える。また、ＯＳ監視ＣＴＲＬ２０３４は、汎用ＯＳ２０１の機能によってリソースへアクセスする操作を監視するモジュールである。

【００１４】

アクセス権限テーブル２０３５は、図３に示すように、リソース指定情報２０３５１、条件２０３５２、ｎ個のアクセス権情報２０３５３～２０３５ｎをリソース毎に登録可能に構成されている。

リソース指定情報２０３５１は、汎用ＯＳ２０１が管理しているリソースのうち、特定のものを指定するための情報であり、例えば、ファイルの場合はファイル名やファイルＩＤなどの情報が登録される。通信データの場合は、ホスト名，ポート番号，ＩＰアドレスなどが登録され、メモリの場合は、そのオブジェクトを示すオブジェクト名，アドレスなどが登録される。また、外部出力装置の場合は、そのデバイスドライバを示すデバイス名などが登録される。

条件２０３５２は、アクセス権が有効となる条件またはその組み合わせをしめすものであり、例えばユーザ名／ＩＤ，グループ名／ＩＤ，時刻，使用アプリケーションなどが登録される。

アクセス権情報２０３５３～２０３５ｎは、既存環境で定義されていない拡張したアクセス権のうち、指定したリソースに付加した権限を示すものであり、例えば他媒体への移動権限，他媒体へのコピー権限，印刷権限、共有メモリへの読み込み権限（Windowsではクリップボードなど），画面ハードコピー権限，使用アプリケーションの限定（特定アクリケーション以外での使用禁止やメール添付の禁止）などが登録される。

なお、一般的に、リソースへのアクセスは複数のＡＰＩによって行われることがあり、その場合はリソース指定情報はＯＳが管理するＩＤ（ハンドルなど）に変換されることがある。その場合、リソース管理プログラム２０３の内部においては、リソース指定情報とそのＩＤは同一視するようにしている。

【００１５】

このような構成に係るリソース管理プログラム２０３の処理について、図２の①～⑨で示す情報伝達手順に従って説明する。

①アプリケーション２０２１からリソースへのアクセス要求があれば、アクセス制御ＣＴＲＬ２０３３に伝える。

②アクセス制御ＣＴＲＬ２０３３は、アクセス権限チェックを行なう際、必要に応じて、アプリケーション２０２１が保持しているリソースの情報をＡＰＬ監視ＣＴＲＬ２０３２から取得する。

③アクセスを拒否する条件として２通りあるが、第１の条件Ａ（アクセス拒否Ａ）では、上記①のアクセス要求に対してアクセス権限チェックを行なう。権限がない場合、アプリケーション２０２１が発行したＡＰＩの処理を行なわずに、結果としてアクセス違反のエラーを返す。

④第２の条件Ｂ（アクセス拒否Ｂ）では、②のアクセス要求に対してアクセス権限チェックを行なう。権限がなく、かつ、アプリケーション２０２１が発行したＡＰＩの結果としてエラーを返すことができない場合、アプリケーション２０２１が要求したリソースへの処理を行なわずに、リソース管理プログラム２０３が予め用意したダミーのリソースへのアクセス要求に代えて、ＡＰＩの処理を行なう。

その結果、アプリケーション２０２１は要求に成功したように動作するが、実

際には要求したリソースにアクセスできない。

【００１６】

⑤アクセス要求①に対してアクセス権限チェックを行った結果、権限がある場合、アプリケーション２０２１が発行したＡＰＩの処理をそのまま汎用ＯＳ２０１に伝え、その結果をアプリケーション２０２１に返す。

⑥上記⑤の処理によって、ＡＰＩが成功し、かつ、そのＡＰＩによってアプリケーション２０２１がリソースを保持する場合は、ＡＰＬ監視ＣＴＲＬ２０３２に伝える。ＡＰＬ監視ＣＴＲＬ２０３２はアプリケーション２０２１と保持しているリソースの対応を登録する。

アプリケーション２０２１がリソースの解放要求ＡＰＩを発行し、かつそのＡＰＩが成功した場合も、ＡＰＬ監視ＣＴＲＬ２０３２に伝える。ＡＰＬ監視ＣＴＲＬ２０３２はアプリケーション２０２１と保持していたリソースの対応を抹消する。

⑦ＯＳ標準機能の操作によって、リソースへのアクセス要求があれば、アクセス制御ＣＴＲＬ２０３３に伝える。

⑧アクセス要求⑦に対してアクセス権限チェックを行なう。権限がない場合、⑦の操作を無視する。

⑨アクセス要求⑦に対してアクセス権限チェックを行なう。権限がある場合、⑦の操作を汎用ＯＳ２０１に伝える。

【００１７】

図４は、目的とするリソースに対するアクセス権限がある場合に、そのリソースを解放するまでのアプリケーション２０２１、リソース管理プログラム２０３、汎用ＯＳ２０１のやり取りを示したＡＰＩの監視及び制御の第１の基本型（１）のシーケンス図である。

この第１の基本型（１）では、アプリケーション２０２１から目的のリソースへのアクセス要求があった場合（ステップ４０１）、リソース管理プログラム２０３はアクセス権があるかチェックし（ステップ４０２）、アクセス権がある場合（ステップ４０３）、汎用ＯＳ２０１にアプリケーション２０２１が発行したＡＰＩをそのまま伝える。汎用ＯＳ２０１は、ＯＳ本来のＡＰＩ処理を行なう（

ステップ４０４）。

リソース管理プログラム２０３は、ＡＰＩが成功した場合、アプリケーション２０２１がそのリソースを保持しているという情報を登録する（ステップ４０５）。そして、汎用ＯＳ２０１からのＡＰＩ結果をそのままアプリケーション２０２１に返す（ステップ４０６）。これにより、リソースへのアクセス完了となる（ステップ４０８）。

【００１８】

この後、アプリケーション２０２１から保持しているリソースの解放要求が発行された場合（ステップ４０８）、リソース管理プログラム２０３はその解放要求を汎用ＯＳ２０１に伝える。汎用ＯＳ２０１は、ＯＳ本来のＡＰＩ処理を行なう（ステップ４０９）。リソース管理プログラム２０３は、ＡＰＩが成功した場合、アプリケーション２０２１がそのリソースを保持しているという情報を解除する（ステップ４１０）。そして、汎用ＯＳ２０１からのＡＰＩ結果をそのままアプリケーション２０２１返す（ステップ４１１）。これにより、保持しているリソースの解放完了となる（ステップ４１２）。

【００１９】

図５は、目的とするリソースに対するアクセス権限がなかった場合に、そのアクセスが拒否されるまでのアプリケーション２０２１、リソース管理プログラム２０３、汎用ＯＳ２０１のやり取りを示したＡＰＩの監視及び制御の第２の基本型（２１）のシーケンス図である。

この第２の基本型（２）では、アプリケーション２０２１から目的のリソースへのアクセス要求があった場合（ステップ５０１）、リソース管理プログラム２０３はアクセス権があるかチェックし（ステップ５０２）、アクセス権がなかった場合（ステップ５０３）、アクセス違反エラーをアプリケーション２０２１に返す（ステップ５０４）。これにより、リソースへのアクセス完了となる（ステップ５０５）。

また、アプリケーション２０２１がアクセス違反エラーに対応していないものにあっては、アプリケーション２０２１から目的のリソースへのアクセス要求があった場合（ステップ５０６）、リソース管理プログラム２０３はアクセス権が

あるかチェックし（ステップ５０７）、アクセス権がなく、かつ、アプリケーション２０２１がアクセス違反エラーに対応していない場合（ステップ５０８）、リソース管理プログラム２０３が予め用意したダミーのリソースへのアクセス要求に置き換え、汎用ＯＳ２０１に渡す（ステップ５０９）。

【００２０】

汎用ＯＳ２０１は、ＯＳ本来のＡＰＩ処理を行なう（ステップ５１０）。リソース管理プログラム２０３は、汎用ＯＳ２０１からのＡＰＩ結果をそのままアプリケーション２０２１返す（ステップ５１１）。この結果、目的のリソースへのアクセス完了となるが、ダミーリソースのため、実質的には何も行われない。

【００２１】

本発明は、以上のようにしてアクセス権限のないリソースへのアクセスを制限するものであるが、汎用のＯＳであるWindowsとＵＮＩＸの場合のＡＰＩを例に挙げて説明する。

まず、ファイルへの複製を禁止する例について説明する。

ファイルへの複製については、従来、読み込み許可ファイルはファイルのコピーが可能であり、その結果オリジナルが複数存在したり、別媒体にて持ち出すことが可能であった。本発明では、ファイルコピーを実現するＡＰＩを監視／制御することにより、権限のないファイルのコピーを禁止する。その場合に、Windowsにおいて監視／制御するＡＰＩとして次のものがある。なお、以下で例示するＡＰＩの機能については、各種の文献で公開されているので、その詳細な説明は省略する。

（１）ファイルオープン／作成ＡＰＩ

CreateFileA

CreateFileW

OpenFile

#lopen

#lcreat

GetOpenFileNameA

GetOpenFileNameW

GetSaveFileNameA

GetSaveFileNameW

（２）ファイルクローズＡＰＩ

CloseHandle

#lclose

（３）ファイルコピー／移動ＡＰＩ

CopyFileA

CopyFileW

MoveFileA

MoveFileW

MoveFileExA

MoveFileExW

DeleteFileA

DeleteFileW

DragQueryFileA

DragQueryFileW

ＵＮＩＸの場合、監視／制御するＡＰＩとしては次のものがある。

（１）ファイルオープン／作成ＡＰＩ

open

creat

（２）ファイルクローズＡＰＩ

close

（３）ファイルコピー／移動ＡＰＩ

rename

【００２２】

このようなＡＰＩの監視によってファイルへの複製を禁止する場合、具体的な方法として３つの方法がある。

＜方法１＞（ファイルオープン中に複製処理を行なうことが判明している場合）

アプリケーションが、複製権限のないファイルをオープンし保持している間（ファイルをクローズするまでの期間）、そのアプリケーションが別のファイルを作成することを拒否する。

＜方法２＞（ファイルクローズ後に複製処理を行なう可能性はあるが、複数ファイルを扱わないことが判明している場合）

アプリケーションが、複製権限のないファイルを一度でもオープンした場合、そのアプリケーションが終了するか、複製権限のあるファイルをオープンするまで、そのアプリケーションが別のファイルを作成することを拒否する。

＜方法３＞（ファイルクローズ後に複製処理を行なう可能性があり、複数ファイルを扱う可能性がある場合）

アプリケーションが、複製権限のないファイルを一度でもオープンした場合、そのアプリケーションが終了するまで、そのアプリケーションが別のファイルを作成することを拒否する。

なお、いずれの方法であっても、別に作成されるファイルによって複製が残ることがないと判明している場合（一時ファイルなどの作成）は拒否しない。

【００２３】

次に、特定ファイルまたは全ての印刷を禁止する例について説明する。従来、印刷機能を実装したアプリケーションによって、ファイルの内容を印刷し、外部に持ち出すことは可能であった。本発明では、印刷を実現するＡＰＩを監視／制御することにより、権限のないファイルの印刷を禁止する。またＦＡＸなどその他の外部出力装置についても、それぞれのデバイスを監視／制御することにより、同様に禁止する。その場合に、Windows及びＵＮＩＸにおいて監視／制御するＡＰＩとして次のものがある。

Windowsの場合

（１）デバイスオープンＡＰＩ

CreateDCA, CreateDCW

（２）デバイスクローズＡＰＩ

ReleaseDC, ClosePrinter

（３）プリンタ選択／ＡＰＬ処理ＡＰＩ

OpenPrinterA

OpenPrinterW

GetPrinterA

GetPrinterW

SetPrinterA

SetPrinterW

SendMessageA

SendMessageW

PostMessageA

PostMessageW

ＵＮＩＸの場合

（１）デバイスオープンＡＰＩ

open

（２）デバイス制御ＡＰＩ

ioctrl

（３）デバイスクローズＡＰＩ

close

【００２４】

このようなＡＰＩの監視によって印刷を禁止する場合、具体的な方法として３つの方法がある。

＜方法１＞（ファイルオープン中に印刷処理可能なことが判明している場合）

アプリケーションが、印刷権限のないファイルをオープンし保持している間（ファイルをクローズするまでの期間）、そのアプリケーションのプリンタ選択、およびプリンタデバイスのオープンを拒否する。

＜方法２＞（ファイルクローズ後に印刷処理を行なう可能性はあるが、複数ファイルを扱わないことが判明している場合）

アプリケーションが、印刷権限のないファイルを一度でもオープンした場合、そのアプリケーションが終了するか、印刷権限のあるファイルをオープンするまで、そのアプリケーションのプリンタ選択、およびプリンタデバイスのオープン

を拒否する。

＜方法３＞（ファイルクローズ後に印刷処理を行なう可能性があり、複数ファイルを扱う可能性がある場合）

アプリケーションが、印刷権限のないファイルを一度でもオープンした場合、そのアプリケーションが終了するまで、そのアプリケーションのプリンタ選択、およびプリンタデバイスのオープンを拒否する。

【００２５】

次に、外部装置の利用を禁止する例について説明する。

従来、ＯＳに装備されている機能や外部装置そのものに権限を付加することは、一般的にはできなかった。本発明では、監視／制御すべきＡＰＩを限定できる機能の指定や、外部装置利用の指定することにより、その利用を禁止する。その場合に、Windows及びＵＮＩＸにおいて監視／制御するＡＰＩとして次のものがある。

Windowsの場合

（１）デバイスオープンＡＰＩ

CreateFileA

CreateFileW

OpenFile

#lopen

#lcreat

（２）デバイスクローズＡＰＩ

CloseHandle

#lclose

ＵＮＩＸの場合

（１）デバイスオープンＡＰＩ

open

（２）デバイス制御ＡＰＩ

ioctrl

（３）デバイスクローズＡＰＩ

close

【００２６】

このようなＡＰＩの監視によって印刷を禁止する場合、具体的な方法として次の方法がある。

＜方法＞

アクセス権限テーブルにて、特定の条件のもとに特定デバイスの使用を禁止されている場合、そのデバイスの利用を以下の方法で拒否する。そのデバイスのデバイス名をもってデバイスオープンＡＰＩ要求があった場合、アクセス禁止エラー、もしくはデバイスが存在しないというエラーを返すことで要求を拒否する。

【００２７】

次に、ファイル内の一部のデータまたは全ての複写を禁止する例について説明する。

従来、アプリケーションによってファイルを画面表示した結果、その内容のすべてまたは一部をＯＳの機能によって複写またはオブジェクトをいう単位で別ファイルに埋め込むことが可能であった。

本発明では、転写や埋め込み機能を実現するＡＰＩ（クリップボードのＡＰＩ、ＯＬＥのＡＰＩなど）を監視／制御することで、権限のない流用を禁止する。その場合に、Windowsにおいて監視／制御するＡＰＩとして次のものがある。

Windowsの場合

（１）複写／埋め込みＡＰＩ

OpenClipboard

SetClipboardData

GetClipboardData

GetOpenClipboardWindow

OleCreate

OleCreateEx

OleCreateFromFile

OleCreateFromFileEx

OleCreateFromData

OleCreateFromDataEx

OleCreateLink

OleCreateLinkEx

OleCreateLinkFromData

OleCreateLinkFromDataEx

OleCreateLinkToFile

OleCreateLinkToFileEx

CloseClipboard

【００２８】

このようなＡＰＩの監視によって複写を禁止する場合、具体的な方法として４つの方法がある。

＜方法１＞（ファイルオープン中に複写処理可能なことが判明している場合）

アプリケーションが、複写権限のないファイルをオープンし保持している間（ファイルをクローズするまでの期間）、そのアプリケーションが複写／埋め込みオブジェクトの形式でデータを登録する際に、拒否もしくは空データを登録する。

＜方法２＞（ファイルクローズ後に複写処理を行なう可能性はあるが、複数ファイルを扱わないことが判明している場合）

アプリケーションが、複写権限のないファイルを一度でもオープンした場合、そのアプリケーションが終了するか、複写権限のあるファイルをオープンするまで、そのアプリケーションが複写／埋め込みオブジェクトの形式でデータを登録する際に、拒否もしくは空データを登録する。

＜方法３＞（ファイルクローズ後に複写処理を行なう可能性があり、複数ファイルを扱う可能性がある場合）

アプリケーションが、複写権限のないファイルを一度でもオープンした場合、そのアプリケーションが終了するまで、そのアプリケーションが複写／埋め込みオブジェクトの形式でデータを登録する際に、拒否もしくは空データを登録する。

＜方法４＞（複写権限のないファイルを埋め込みオブジェクトとして取り込む

場合）

複写権限のないファイルを取り込む処理を行なう際に、オブジェクトの登録もしくはそのオブジェクトの取得ＡＰＩにおいて、アクセス違反のエラーを返すか、空データを登録あるいは取得することで、処理要求を拒否する。

【００２９】

次に、ネットワークを介してファイルが外部へ流出することを禁止する例について説明する。

従来、ファイルコピー以外に、ＦＴＰプログラムのように、ネットワークを介してファイルを外部へ転送することは可能であった。本発明では、ネットワークリソースにアクセスするＡＰＩを監視／制御することで、権限のないファイルを使用中のアプリケーションから外部への流出を禁止する。その場合に、Windows及びＵＮＩＸにおいて監視／制御するＡＰＩとして次のものがある。

（１）Windowsの場合

WSAStartup

accept

bind

connect

gethostbyname

gethostbyaddr

getprotobyname

getprotobynumber

getservbyname

getservbyport

getpeername

getsockname

gethostname

getsockopt

setsockopt

recv

recvfrom

socket

select

send

sendto

WSASend

WSASendTo

WSAAsyncSelect

WSAAsyncGetHostByAddr

WSAAsyncGetHostByName

WSAAsyncGetProtoByNumber

WSAAsyncGetProtoByName

WSAAsyncGetServByPort

WSAAsyncGetServByName

WSACancelAsyncRequest

WSASetBlockingHook

WSAUnhookBlockingHook

WSACleanup

closesocket

shutdown

（２）ＵＮＩＸの場合

accept

bind

connect

gethostbyname

gethostbyaddr

getprotobyname

getprotobynumber

getservbyname

getservbyport

getpeername

getsockname

gethostname

getsockopt

setsockopt

recv

recvfrom

socket

select

send

sendto

closesocket

shutdown

【００３０】

このようなＡＰＩの監視によって外部への流出を禁止する場合、具体的な方法として３つの方法がある。

＜方法１＞（ファイルオープン中に出力処理可能なことが判明している場合）

アプリケーションが、外部出力権限のないファイルをオープンし保持している間（ファイルをクローズするまでの期間）、そのアプリケーションからの接続要求や送信要求を、アクセス違反もしくはタイムアウトなどのエラーで拒否する。

＜方法２＞（ファイルクローズ後に出力処理を行なう可能性はあるが、複数ファイルを扱わないことが判明している場合）

アプリケーションが、出力権限のないファイルを一度でもオープンした場合、そのアプリケーションが終了するか、出力権限のあるファイルをオープンするまで、そのアプリケーションの接続要求や送信要求を、アクセス違反もしくはタイムアウトなどのエラーで拒否する。

＜方法３＞（ファイルクローズ後に出力処理を行なう可能性があり、複数ファイルを扱う可能性がある場合）

アプリケーションが、出力権限のないファイルを一度でもオープンした場合、そのアプリケーションが終了するまで、そのアプリケーションの接続要求や送信要求を、アクセス違反もしくはタイムアウトなどのエラーで拒否する。

ただし、その通信によってデータ出力されないことが判明している場合は、拒否しない。

【００３１】

次に、ファイルの内容のイメージを取得することを禁止する例について説明する。

ＯＳの機能として画面全体や一部、またはウインドウ単位のハードコピーをイメージデータとして取得することが、一般的には可能であり、従来、そのイメージデータを流用、持ち出しすることができた。本発明では、画面内のイメージデータ取得ＡＰＩを監視／制御することで、イメージデータ取得を禁止する。

その場合に、Windowsにおいて監視／制御するＡＰＩとして次のものがある。

（１）デバイスオープンＡＰＩ

GetWindowDC

WindowFromDC

GetDC

GetDCEX

GetDesktopWindow

GetDeviceCaps

CreateDCA

CreateDCW

（２）イメージ取得ＡＰＩ

BitBlt

StretchBlt

（３）デバイスクローズＡＰＩ

DeleteDC

ReleaseDC

【００３２】

このようなＡＰＩの監視によってイメージを取得することを禁止する場合、具体的な方法として３つの方法がある。

＜方法１＞（画面全体のハードコピーを拒否する場合）

現在画面上に表示されているウインドウを所有しているアプリケーションが、ハードコピー取得権限のないファイルを保持している場合、画面全体のハードコピー取得を拒否する。画面全体のハードコピーは、画面全体を管理しているウインドウ（Windowsの場合はデスクトップウインドウ）を監視することで、＜方法２＞と同じ。

WindowsにおけるDirectDrawなど、画面全体のハードコピーを取得するＡＰＩが存在すれば、同様に拒否する。

さらに、ディスプレイデバイスからＶＲＡＭイメージを取得するアプリケーションに対しては、それを拒否する。

＜方法２＞（ウインドウのハードコピーを拒否する場合）

現在画面上に表示されているウインドウを所有しているアプリケーションが、ハードコピー取得権限のないファイルを保持している場合、そのウインドウのハードコピー取得を拒否する。ウインドウが画面上に表示されているかは、ウインドウの状態を監視することで行なう。

また、ハードコピー取得の拒否は、そのウインドウに関連付けられたデバイスコンテキストからのイメージコピーを拒否することで行なう。

＜方法３＞（画面の一部のハードコピーを拒否する場合）

ハードコピーを取得する領域が判断できる場合は、＜方法１＞における条件を、取得領域が対象となるウインドウと重なる時とし、以降は＜方法１＞と同じ。また領域が判断できない時は、＜方法１＞と同じ。

【００３３】

次に、ファイル種別毎に利用アプリケーションを限定する例について説明する。

従来、アプリケーション利用に制限がないため、参照以外の目的でファイルにアクセス可能であった。本発明では、ファイルごとに利用アプリケーションを限定できる。その場合に、Windowsにおいて監視／制御するＡＰＩとして次のものが

ある。

（１）ファイルオープンＡＰＩ

CreateFileA

CreateFileW

OpenFile

#lopen

#lcreat

（２）ファイルクローズＡＰＩ

CloseHandle

#lclose

（３）プロセス管理ＡＰＩ

WinExec

CreateProcessA

CreateProcessW

ExitProcess

ＵＮＩＸの場合

（１）ファイルオープンＡＰＩ

open

（２）ファイルクローズＡＰＩ

close

【００３４】

このようなＡＰＩの監視によって利用アプリケーションを限定する場合、具体的な方法として次の方法がある。

＜方法＞

アプリケーションがファイルをオープンする際、そのファイルの権限をチェックし、許可されたアプリケーションでない場合はアクセス違反エラーを返すことで、オープン要求を拒否する。

次に、ＯＳの特定の機能の利用を禁止する例について説明する。

従来、ＯＳに装備されている機能に権限を付加することは、一般的にはできな

かった。本発明では、監視／制御すべきＡＰＩを限定する機能を指定することで、その利用を禁止することができる。例えば、ファイルのタイムスタンプやシステム日時の変更を禁止するなどである。その場合に、Windowsにおいて監視／制御するＡＰＩとして次のものがある。

（１）ファイルのタイムスタンプ変更ＡＰＩ

SetFileTime

（２）システム日時の変更ＡＰＩ

SetSystemTime

SetSystemTimeAdjustment

【００３５】

このようなＡＰＩの監視によってＯＳにおける特定の機能の利用を禁止する場合、具体的な方法として次の方法がある。

＜方法＞

特定の条件下で禁止されているＡＰＩが発行された際に、アクセス違反エラーを返すか、実際の処理を行なわずに（ダミー処理）正常リターンすることで、禁止されているＡＰＩ（ＯＳ機能）を拒否する。

【００３６】

次に、プロセス内メモリの参照または変更を禁止する例について説明する。

従来、アプリケーションが明示的に拒否しない限り、プロセス内メモリの参照／変更を禁止することができなかった。本発明では、プロセス内メモリの参照／変更ＡＰＩを監視／制御することで、他のアプリケーションからの参照／変更を禁止することができる。

その場合に、Windowsにおいて監視／制御するＡＰＩとして次のものがある。

（１）プロセス管理ＡＰＩ

OpenProcess

CreateProcess

CloseHandle

（２）メモリ操作ＡＰＩ

ReadProcessMemory

WriteProcessMemory

ReadProcessMemoryVlm

WriteProcessMemoryVlm

【００３７】

このようなＡＰＩの監視によってプロセス内メモリの参照または変更を禁止する印刷を禁止する場合、具体的な方法として次の方法がある。

＜方法＞

アクセスが禁止されているアプリケーションのプロセス内メモリにおいて、メモリ操作ＡＰＩが要求された際に、アクセス違反エラーを返す。

次に、ブラウザに表示したＷｅｂページの印刷や保存や外部装置への出力を禁止する例について説明する。

従来、閲覧や再生のみを許可したＷｅｂページでも、実際にはブラウザソフトによって印刷や保存が可能であった。ＷｅｂページをロードするためのネットワークリソースにアクセスするＡＰＩを監視し、ブラウザが行う印刷や保存のアプリケーションを監視／制御することにより、印刷や保存や外部装置への出力操作を禁止することができる。その場合に、監視／制御するＡＰＩとして次のようなものがある。

Windowsの場合

（１）通信ＡＰＩ

WSAStartup

accept

bind

connect

gethostbyname

gethostbyaddr

getprotobyname

getprotobynumber

getservbyname

getservbyport

getpeername

getsockname

gethostname

getsockopt

setsockopt

recv

recvfrom

socket

select

send

sendto

WSASend

WSASendTo

WSAAsyncSelect

WSAAsyncGetHostByAddr

WSAAsyncGetHostByName

WSAAsyncGetProtoByNumber

WSAAsyncGetProtoByName

WSAAsyncGetServByPort

WSAAsyncGetServByName

WSACancelAsyncRequest

WSASetBlockingHook

WSAUnhookBlockingHook

WSACleanup

closesocket

shutdown

（２）その他、前述のファイル、印刷、外部装置への操作を禁止する場合のＡＰＩ

UNIXの場合

（１）

accept

bind

connect

gethostbyname

gethostbyaddr

getprotobyname

getprotobynumber

getservbyname

getservbyport

getpeername

getsockname

gethostname

getsockopt

setsockopt

recv

recvfrom

socket

select

send

sendto

closesocket

shutdown

（２）その他、前述のファイル、印刷、外部装置への操作を禁止する場合のＡＰＩ

【００３８】

このような通信ＡＰＩを監視し、印刷や保存や外部装置への出力を禁止する方法として、次の方法がある。

まず、Ｗｅｂページ内に記述された禁止指定を読み取る。具体的には、ｈｔｔ

ｐプロトコルまたは同等のプロトコルのデータを監視し、その中のＷｅｂページデータ部分に印刷や保存の禁止指定タグが含まれていれば、そのＷｅｂページは印刷や保存が禁止されていると判断する。または、権限の獲得を利用者に求め、獲得できなかった場合に印刷や保存が禁止されていると判断する。しかし、獲得できた場合には、印刷や保存が禁止されていないものと判断する。すなわち、アクセス権限が獲得できたか否かをもって、アクセス権限があるか否かを判定する。

印刷や保存や外部装置への出力が禁止されているページを表示しているブラウザが、印刷や保存を行なおうとした場合、前述した印刷やファイルの保存や外部装置への出力を禁止する方法を用いて、それを禁止する。

ここで説明したＷｅｂページの例は、そのしくみの類似性により、容易にディジタルテレビジョンや携帯電話機、携帯型情報端末等のコンテンツにおいても利用できるものである。

【００３９】

次に、リソース管理プログラムを応用した例を示しておく。

図６は、リソース管理プログラム２０３が管理しているリソースのアクセス状況を履歴管理プログラム６０１に転送し、履歴管理データベース（ＤＢ）６０２に格納しておき、必要に応じて、図８に示すようなアクセス監視履歴として画面表示する構成を示したものである。通報プログラム６０３は、不正なアクセスがあった場合にシステム管理者の端末に対し、図６（ｂ）で示すような内容の不正アクセス通知画面を送信し、表示させるものである。

なお、一般ユーザが不正アクセスを行なった場合には、図６（ａ）で示すような画面表示が行われる。

【００４０】

なお、上記の説明においては、アクセス権管理テーブルを参照してアクセス権限の有無を判定するようにしているが、コンピュータリソース内部に記述された、既存環境で定義されていない拡張したアクセス権を指定するアクセス権限情報を参照し、アクセス権限があるか否かを判定するようにすることもできる。

また、上記の説明において用いたネットワークリソースとは、通信媒体、デバ

イス、アクセスポイント、ディジタルテレビジョンのチャンネル、通信データまたはコンテンツなど、ＯＳが管理しているリソースのうちネットワークに関するものである。

【００４１】

以上のように、本実施形態においては、基本的には、ファイル、ネットワーク、記憶装置、表示画面、外部付属装置等のオペレーティングシステムが管理しているコンピュータリソースに対するプロセスまたはオペレーティングシステムからの操作要求をコンピュータリソースにアクセスする前に捕捉し、その捕捉した操作要求によって指定されるコンピュータリソースに対するアクセス権限があるか否かを判定し、アクセス権限があれば当該操作要求通りにオペレーティングシステムに渡し、その結果を要求元プロセスに返し、アクセス権限がなければ当該操作要求を拒否するようにしたため、ＯＳやプロセス（ＯＳの元に稼動しているプログラムであり、アプリケーションやデーモンなど）を変更することなく、ファイルや画面以外のコンピュータリソースを含めてアクセス権限のないユーザに対するリソースの操作を制限することができる。

また、リソース管理プログラムを既存の環境に組み込むだけで、上述したような各種の不正アクセスを制限することができ、既存のアクセス権の範囲を拡張することが可能になる。

【００４２】

さらに、要求元のアプリケーションがアクセス違反に対応する機能を有していない場合であっても、ダミーのコンピュータリソースへの操作要求に変換してオペレーティングシステムに渡すようにしたため、アクセス違反に対応する機能を有していないアプリケーションに対しても対応することができる。

【００４３】　なお、リソース管理プログラムは、ＣＤ－ＲＯＭ等のディスク型ストレージ、半導体メモリ及び通信ネットワークなどの各種の媒体を通じてコンピュータにインストールまたはロードすることができる。また、プログラム製品単体として、コンピュータユーザに提供することができる。

また、実施形態で例示したＡＰＩについては、その一例を示しただけであって、ＯＳのバージョンアップなどによって追加された場合でも容易に対応できるこ

とは言うまでもない。

【００４４】

【発明の効果】

以上の説明から明らかなように、本発明は、基本的には、ファイル、ネットワーク、記憶装置、表示画面、外部付属装置等のオペレーティングシステムが管理しているコンピュータリソースに対するプロセスまたはオペレーティングシステムからの操作要求をコンピュータリソースにアクセスする前に捕捉し、その捕捉した操作要求によって指定されるコンピュータリソースに対するアクセス権限があるか否かを判定し、アクセス権限があれば当該操作要求通りにオペレーティングシステムに渡し、その結果を要求元プロセスに返し、アクセス権限がなければ当該操作要求を拒否するようにしたため、ＯＳやプロセス（ＯＳの元に稼動しているプログラムであり、アプリケーションやデーモンなど）を変更することなく、ファイルや画面以外のコンピュータリソースを含めてアクセス権限のないユーザに対するリソースの操作を制限することができる。

また、リソース管理プログラムを既存の環境に組み込むだけで、上述したような各種の不正アクセスを制限することができ、既存のアクセス権の範囲を拡張することが可能になる。

また、リソース制御プログラムを既存の環境に組み込むだけで、各種の不正アクセスを制限することができ、従来のアクセス権の範囲を拡張することが可能になる。

さらに、アクセス違反に対応する機能を有していないアプリケーションに対しても対応することができるなどの効果が得られる。

【図面の簡単な説明】

【図１】

本発明の実施環境の一実施形態を示すハードウェア構成図である。

【図２】

本発明に係るリソース管理プログラムの機能構成及びＯＳとアプリケーションとの関係を示す図である。

【図３】

アクセス管理テーブルのデータ構成例を示す図である。

【図４】

本発明におけるＡＰＩの監視／制御の第１の基本型を示すシーケンス図である。

【図５】

本発明におけるＡＰＩの監視／制御の第２の基本型を示すシーケンス図である。

【図６】

アクセス違反があった場合に表示される画面の例を示す図である。

【図７】

アクセス履歴を記録し、不正アクセスを通知する機能を加えた実施形態を示す図である。

【図８】

アクセス監視履歴の表示画面の例を示す図である。

図である。

【図９】

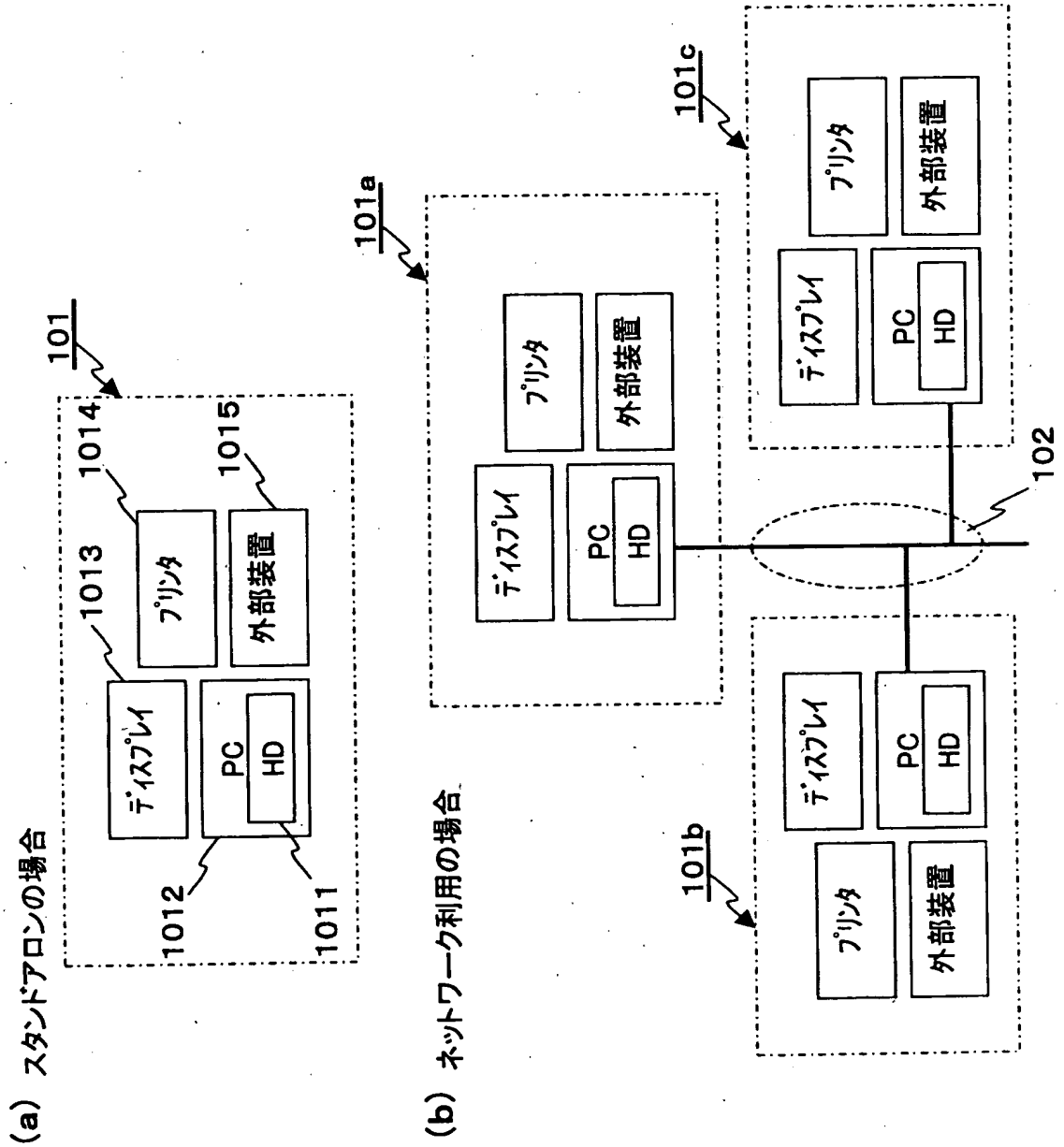
アクセス制限対象となるリソースの例を示す図である。

【符号の説明】

１０１…コンピュータ、１０２…ネットワーク、２０１…汎用ＯＳ、２０３…リソース管理プログラム、６０１…履歴管理プログラム、６０３…通報プログラム、２０３１…ＡＰＩ監視コントローラ、２０３２…ＡＰＬ監視コントローラ、２０３３…アクセス制御コントローラ、２０３４…ＯＳ監視コントローラ、２０３５…アクセス権管理テーブル。

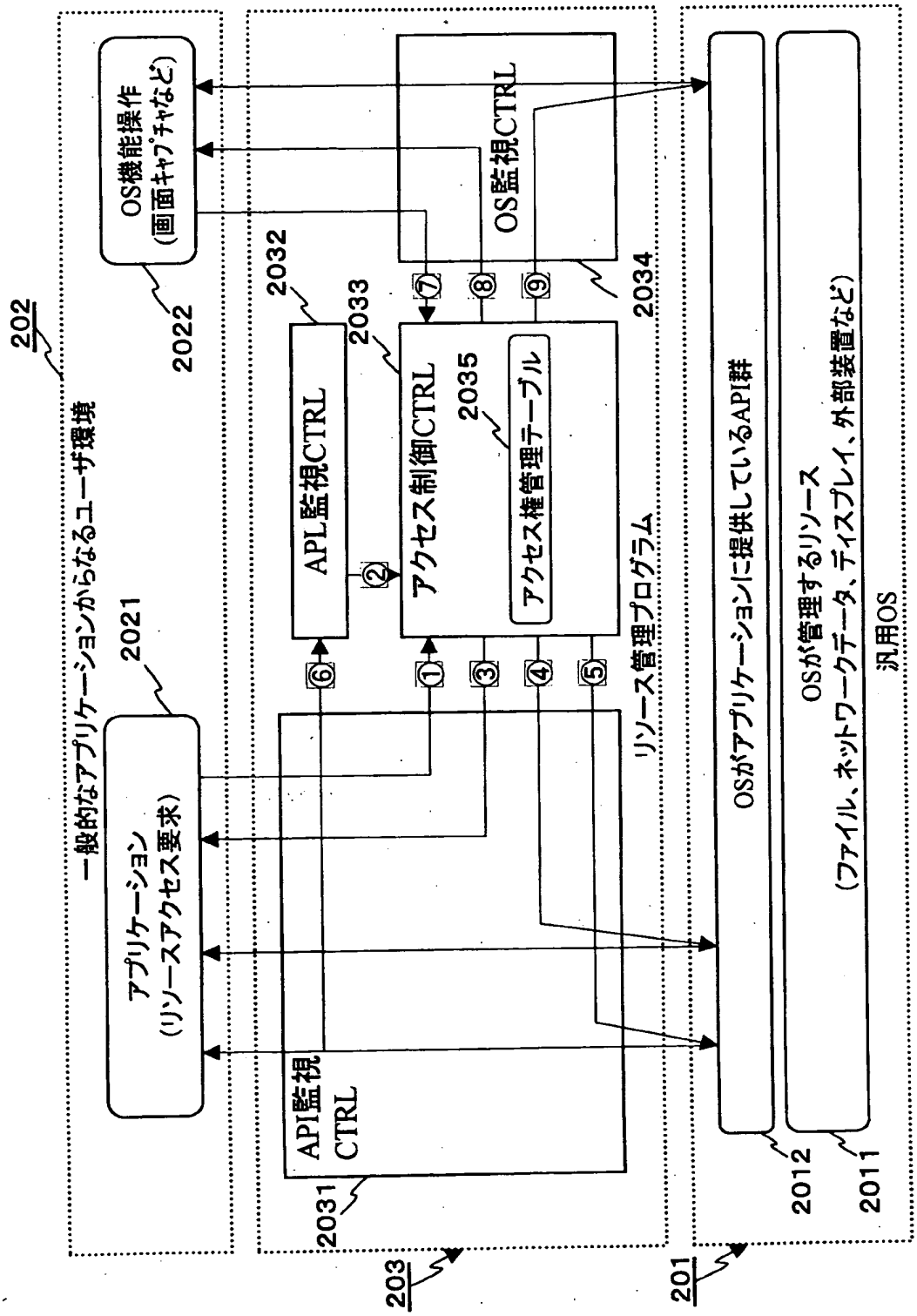
【書類名】　　図面

【図１】

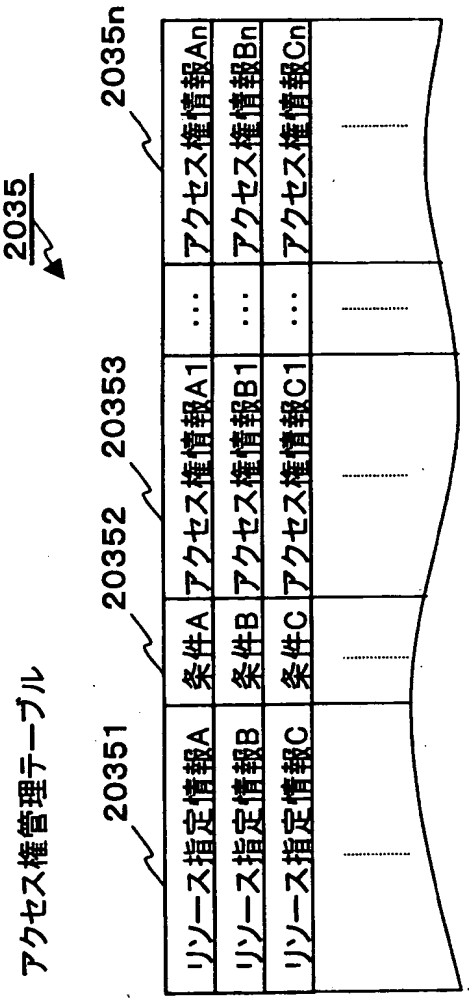


出証特２００１－３０９８２８７

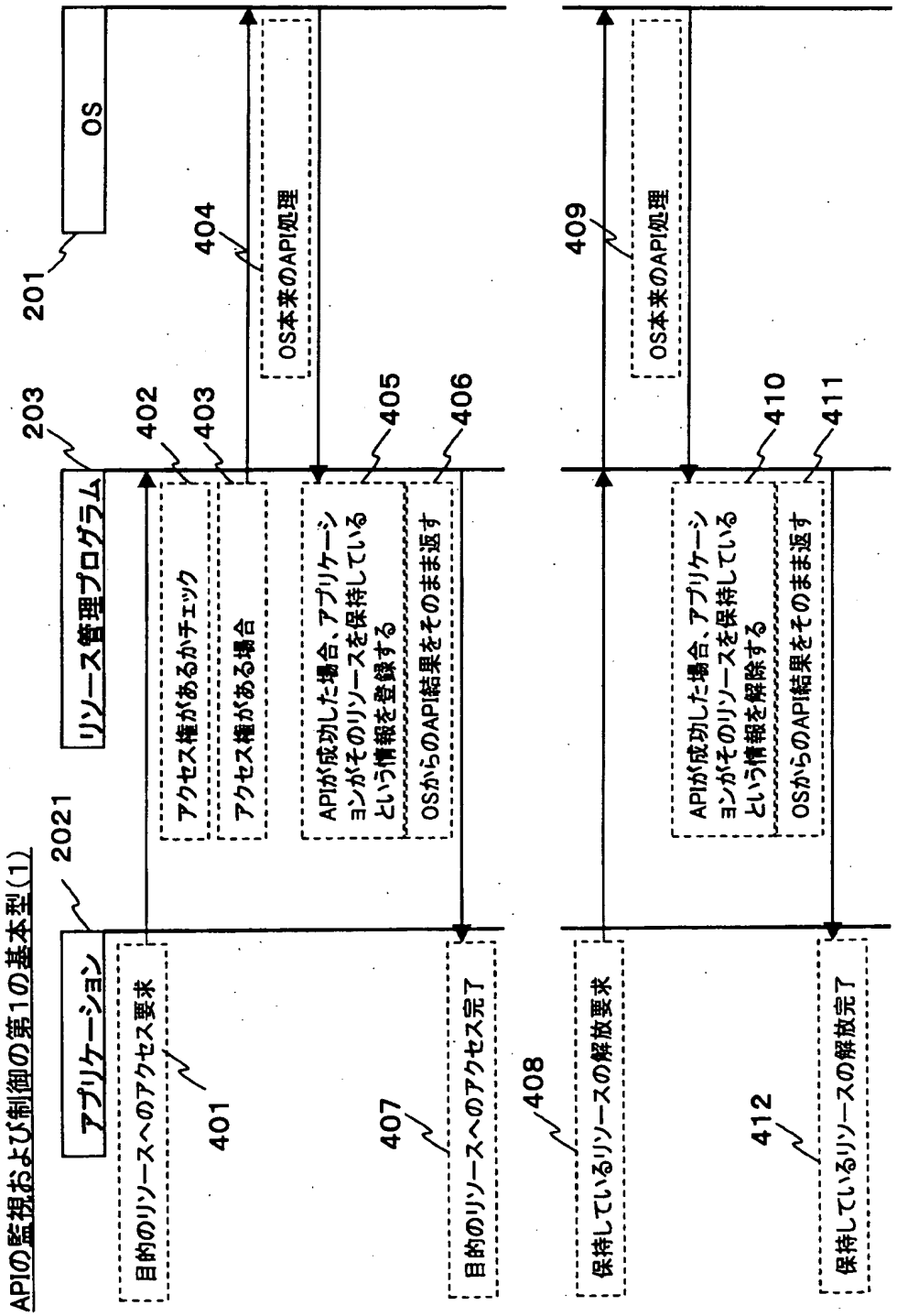
【図２】


202
一般的なアプリケーションからなるユーザ環境
2021
アプリケーション
(リソースアクセス要求)
2022
OS機能操作
(画面キャプチャなど)
2032
APL監視CTRL
2033
アクセス制御CTRL
2035
アクセス権管理テーブル
2034
OS監視CTRL
①
②
③
④
⑤
⑥
⑦
⑧
⑨
リソース管理プログラム
API監視
CTRL
2031
203
201
OSがアプリケーションに提供しているAPI群
2012
OSが管理するリソース
(ファイル、ネットワークデータ、ディスプレイ、外部装置など)
2011
汎用OS

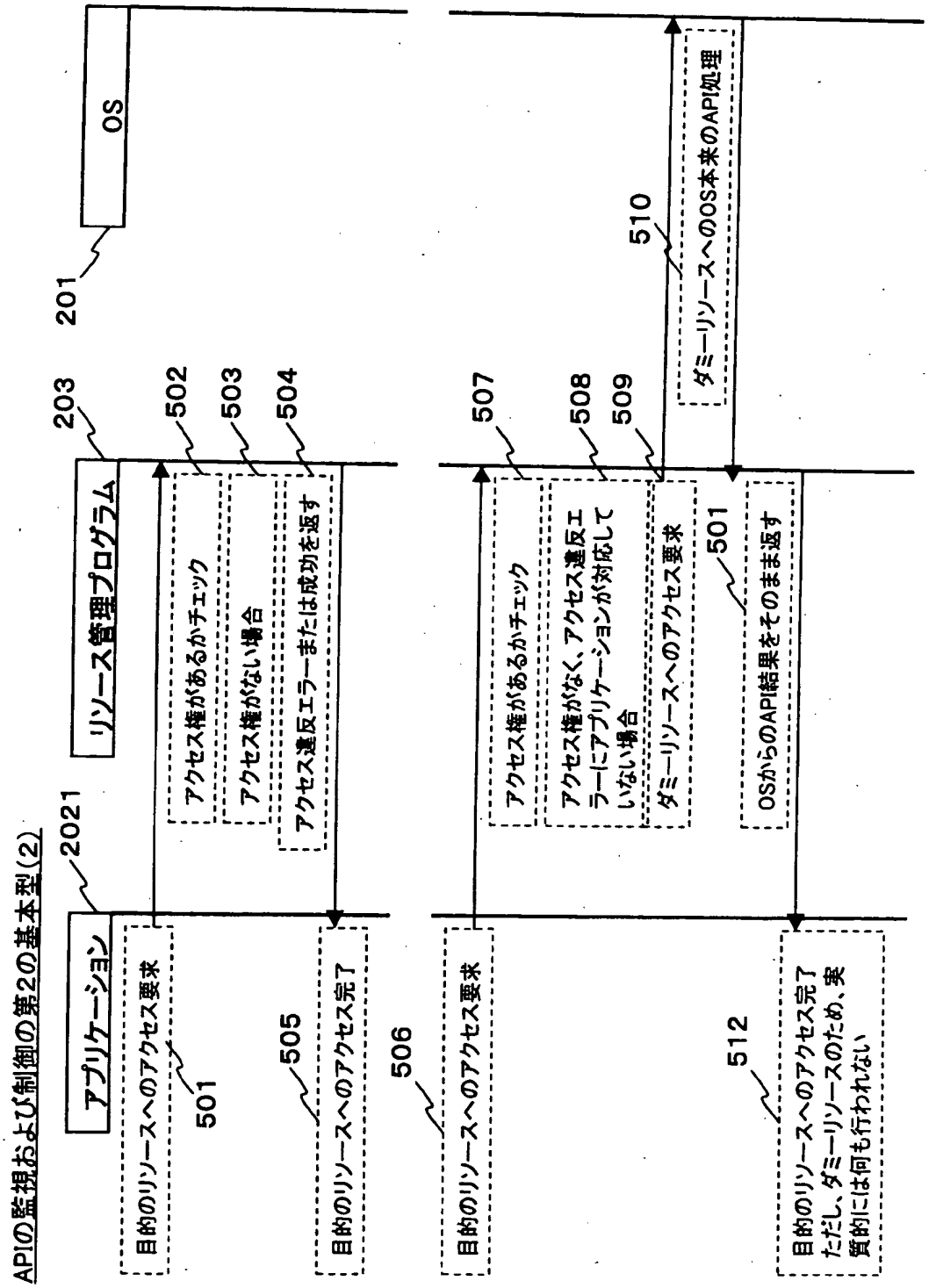
【図３】


アクセス権管理テーブル
2035
20351
20352
20353
2035n
リソース指定情報A
リソース指定情報B
リソース指定情報C
条件A
条件B
条件C
アクセス権情報A1
アクセス権情報B1
アクセス権情報C1
アクセス権情報An
アクセス権情報Bn
アクセス権情報Cn

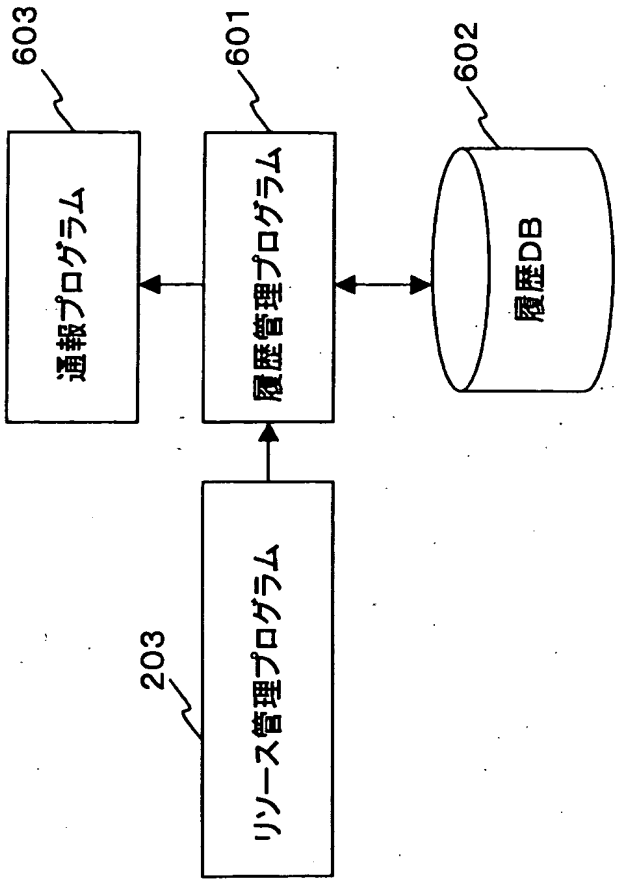
【図４】


APIの監視および制御の第1の基本型(1)
アプリケーション
2021
リソース管理プログラム
203
OS
201
目的のリソースへのアクセス要求
401
アクセス権があるかチェック
402
アクセス権がある場合
403
OS本来のAPI処理
404
APIが成功した場合、アプリケーションがそのリソースを保持しているという情報を登録する
405
OSからのAPI結果をそのまま返す
406
目的のリソースへのアクセス完了
407
保持しているリソースの解放要求
408
OS本来のAPI処理
409
APIが成功した場合、アプリケーションがそのリソースを保持しているという情報を解除する
410
OSからのAPI結果をそのまま返す
411
保持しているリソースの解放完了
412

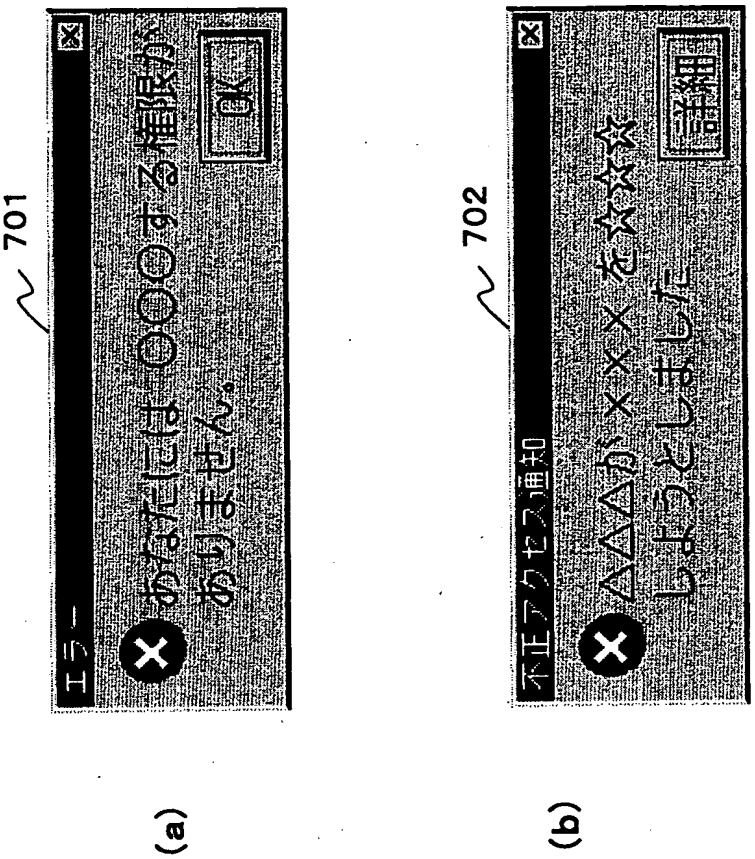
【図５】


APIの監視および制御の第2の基本型(2)
2021
アプリケーション
203
リソース管理プログラム
201
OS
目的のリソースへのアクセス要求
501
アクセス権があるかチェック
502
アクセス権がない場合
503
アクセス違反エラーまたは成功を返す
504
目的のリソースへのアクセス完了
505
目的のリソースへのアクセス要求
506
アクセス権があるかチェック
507
アクセス権がなく、アクセス違反エラーにアプリケーションが対応していない場合
508
ダミーリソースへのアクセス要求
509
ダミーリソースへのOS本来のAPI処理
510
OSからのAPI結果をそのまま返す
501
目的のリソースへのアクセス完了
ただし、ダミーリソースのため、実質的には何も行われない
512

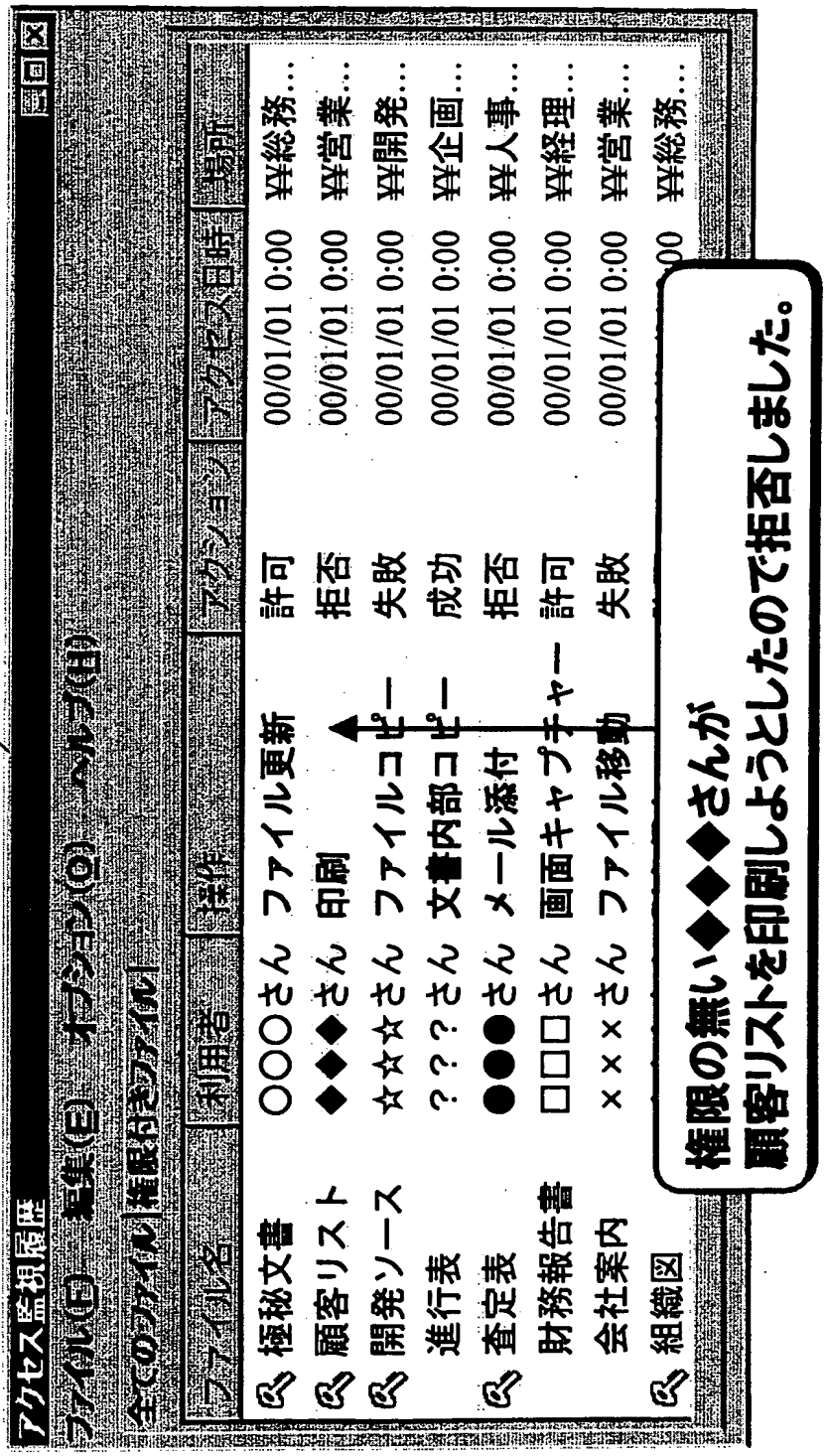
【図６】



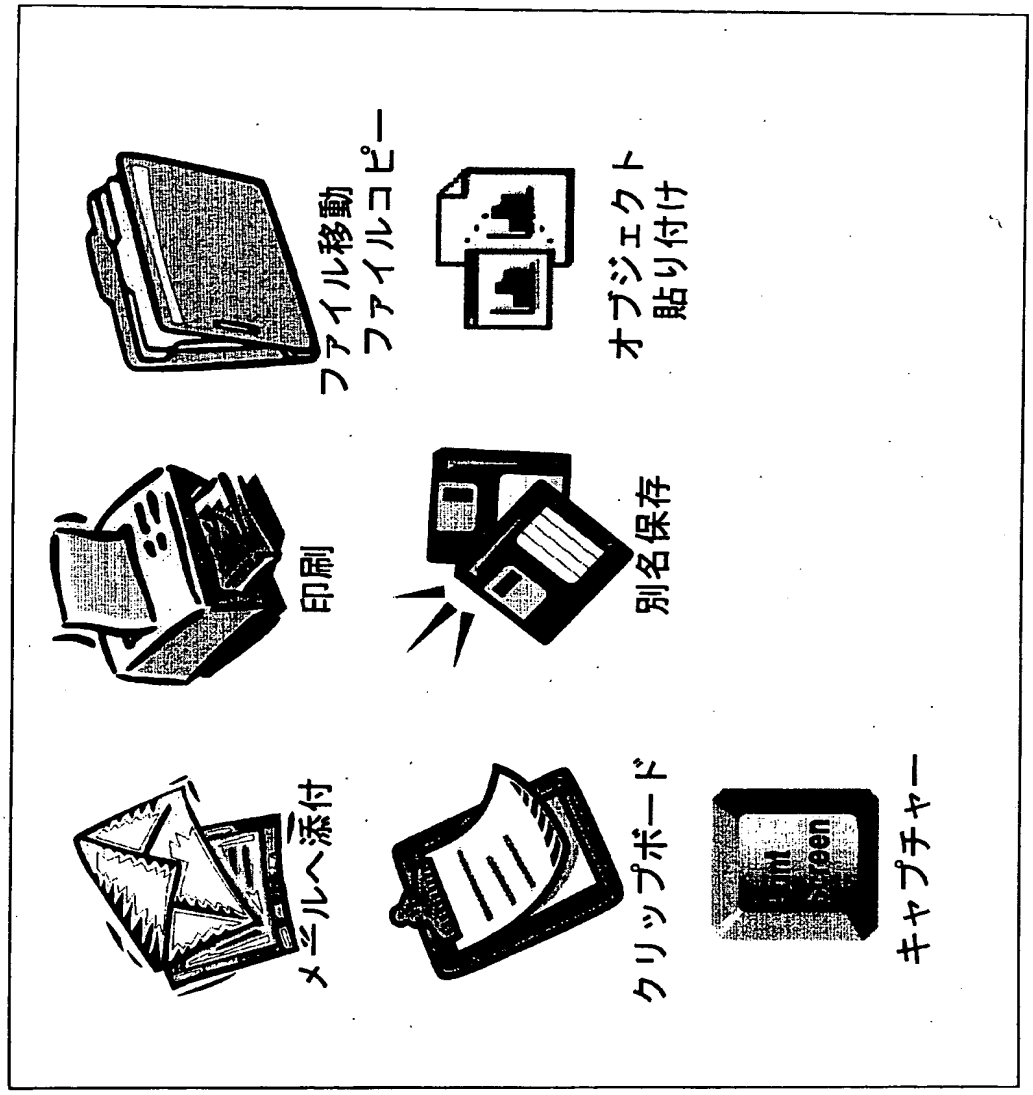
【図７】



【図８】


801
アクセス監視履歴
ファイル(F)　編集(E)　オプション(O)　ヘルプ(H)
全てのファイル　権限付きファイル
ファイル名
利用者
操作
アクション
アクセス日時
場所
極秘文書　○○○さん　ファイル更新　許可　00/01/01 0:00　¥¥総務…
顧客リスト　◆◆◆さん　印刷　拒否　00/01/01 0:00　¥¥営業…
開発ソース　☆☆☆さん　ファイルコピー　失敗　00/01/01 0:00　¥¥開発…
進行表　？？？さん　文書内部コピー　成功　00/01/01 0:00　¥¥企画…
査定表　●●●さん　メール添付　拒否　00/01/01 0:00　¥¥人事…
財務報告書　□□□さん　画面キャプチャー　許可　00/01/01 0:00　¥¥経理…
会社案内　×××さん　ファイル移動　失敗　00/01/01 0:00　¥¥営業…
組織図　00　¥¥総務…
権限の無い◆◆◆さんが
顧客リストを印刷しようとしたので拒否しました。

【図９】



【書類名】　　要約書

【要約】

【課題】　ＯＳやプロセスを変更することなく、アクセス権限のないユーザに対するリソースの操作を制限し、しかも既存環境における禁止または制限事項を拡張すること。

【解決手段】　ファイル、ネットワーク、記憶装置、表示画面、外部付属装置等のオペレーティングシステムが管理しているコンピュータリソースに対するプロセスまたはオペレーティングシステムからの操作要求をコンピュータリソースにアクセスする前に捕捉し、その捕捉した操作要求によって指定されるコンピュータリソースに対するアクセス権限があるか否かを判定し、アクセス権限があれば当該操作要求通りにオペレーティングシステムに渡し、その結果を要求元プロセスに返し、アクセス権限がなければ当該操作要求を拒否する。

【選択図】　　図２

出　願　人　履　歴　情　報

識別番号　　〔５００００８３２２６〕

１．変更年月日　　２０００年　２月２５日
［変更理由］　新規登録
住　所　東京都中央区月島１丁目２番１３号
氏　名　ハミングヘッズ株式会社